HIGH DIMENSIONAL TIME SERIES ANOMALY DETECTION

A Thesis

by

DEVESH KUMAR

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Xia 'Ben' Hu
Committee Members,	Theodora Chaspari
	Na Zou
Head of Department,	Scott Schaefer

August 2021

Major Subject: Computer Science

ABSTRACT

Anomaly Detection task is to determine critical data points whose behaviour deviates unexpectedly from usual data points behaviour. These anomalous data points might indicate a major fault in a manufacturing unit, security glitch on a server, fraud in banking system or abnormal functioning of a human body part. These data are usually recorded using several high precision sensors to capture multiple contributing factors as multivariate time series data.

To determine the anomalous data on such high dimensional real world data using data driven machine learning approaches, it is important to extract temporal information among data points and latent feature based methods are instinctive choice. In this thesis primarily, we will propose an Anomaly Aware Matrix Factorization (ATMF) method with two temporal neighborhood term , the autoregressive bias which could learn local patterns and the moving average bias, which could smoothen noises. To develop an optimization function which will be robust to outliers we will use an approx mean absolute error function. ATMF performances will be demonstrated using 5 real world dataset. This thesis further propose future modifications in this model to encode contextual information within model using a minimum arborescence tree.

In the second part, we will briefly discuss, an automated time series outlier detection System (TODS) package for high dimensional time series data, which has several modules for data preprocessing, feature extractions and anomaly detection.

ACKNOWLEDGMENTS

My deepest gratitude goes to my advisor Dr Xia "Ben" Hu , for accepting me as his student and supporting and guiding my through my graduate education. His deepest insights on the subject matter was a valuable source of learning. Thanks for being constant source of inspiration and encouraging me to remain focused on my research ideas .

I am grateful to Dr Na Zou for letting me work on real world projects and her valuable mentorship and inputs to my research. I like to thank Dr Theodora Chaspari for her supportive gestures and valuable feedbacks on my reseach

I would also like to thank kwei-herng lai for the constant exchange of ideas and providing me with feedback about my research and my Data Lab members for their constant support and encouragement .

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a Thesis Committee consisting of Dr.Xia "Ben" Hu and Dr. Theodora Chaspari from the Department of Computer Science and Engineering, and and Dr. Na Zou from the Department of Engineering Technology and Industrial Distribution.

The experiments for Anomaly Aware Matrix Factorization method was assisted by Kwei-herng lai. The TODS package introduced briefly in this thesis was a combined efforts of members of DATA Lab but the works mentioned are contributed by the student independently .

Funding Sources

This work was not funded by any external sources.

NOMENCLATURE

AD	Anomaly Detection
MSE	Mean Squared Error
MAE	Mean Absolute Error
ATMF	Anomaly Aware Matrix Factorization
TRMF	Temporal Regularized Matrix Factorization
TODS	Time Series Outlier Detection System
AR	Auto Regression
LSTM	Long Term Short Term Memory
LSTM ED	Long Term Short Term Memory Encoder Decoder
AE	Auto Encoder
RNN	Recurrent Neural Network
MA	Moving Average
MF	Matrix Factorization
OCSVM	One Class Support Vector Machine
iForest	Isolation Forest
SVD	Singular Value Decomposition
VAE	Variational Auto Encoder
ND	Normal Deviation
NRMSE	Normal Root Mean Square Error
DAG	Directed Acyclic Graph
DAGMM	Deep Autoencoding Gaussian Mixture Model

TABLE OF CONTENTS

Page

ABSTRACT ii

ACKNOWLEDGMENTS iii

CONTRIBUTORS AND FUNDING SOURCES iv

NOMENCLATURE v

TABLE OF CONTENTS vi

LIST OF FIGURES viii

LIST OF TABLES ix

1. INTRODUCTION 1

2. RELATED WORKS 5

   2.1 Classical Time-Series Models 5
   2.2 Matrix Factorization for High-dimensional Time-series Analysis 6
   2.3 Time Domain and Frequency Domain Features Analyser 7
   2.4 Anomaly Detection 8

3. ANOMALY AWARE MATRIX FACTORIZATION 10

   3.1 Methodology 11
      3.1.1 Temporal Neighborhood Model 11
      3.1.2 Autoregressive Bias 11
      3.1.3 Moving Average Bias 12
      3.1.4 Temporal Biased Matrix Factorization 12
      3.1.5 Anomaly-aware Penalty Function 13
   3.2 Anomaly-aware Temporal Matrix Factorization 15
   3.3 Experiment and Analysis 17
      3.3.1 Experiment Settings and Datasets 17
         3.3.1.1 Time Series Analysis 17
         3.3.1.2 Anomaly Detection 18
         3.3.1.3 Baselines 19
      3.3.2 Classical Time Series Tasks 20
      3.3.3 Anomaly Detection 21
   3.4 Anomaly-aware Temporal Matrix Factorization 22

3.4.1 Case studies on MAE loss and MSE loss for anomaly detection 22
3.5 Future Works 24

4. TODS: TIME SERIES OUTLIER DETECTION SYSTEM -AUGMENTATION 27

5. CONCLUSION.. 30

5.1 Further Studies 30

REFERENCES 31

APPENDIX A. EXPERIMENT DETAILS FOR ATMF 35

A.1 Time Series Classical Tasks 35
A.2 Anomaly Detection 37
A.2.1 Implementation Details 37
A.3 Comprehensive Results for Anomaly Detection Task 38
A.4 Additional analysis... 38

LIST OF FIGURES

FIGURE Page

1.1 The factorization of multivariate time series matrix. A multivariate time series signal can be seen as an n by t matrix. The goal is to preserve the temporal information in k-dimensional latent features X and to model the correlation between variables in k-dimensional latent features F. 3

2.1 An illustration showing the similarity between neighborhood model for item-based collaborative filtering (left) and autoregressive model for time series modeling (right). As the figure shown, both models capture the correlation between the current data point and its neighbors. In the neighborhood model, the neighboring items are used to predict the missing rating of U3 on I4. The autoregressive model leverages the past neighboring points to predict the current time point. 7

3.1 As residual grows, the MSE (square L2 norm) and MAE (L1 norm) assign the penalties with quadratic growth and linear growth, respectively. 16

3.2 The smooth time series of Yahoo-A1 real_3 23

3.3 The noisy time series of Yahoo-A1 real_1 23

3.4 Case studies of MAE and MSE loss functions on noisy data. The MAE loss gives higher prediction error than the MSE loss does, making it more suitable for anomaly detection. 23

3.5 A depiction of time series data as directed acyclic graph (DAG). Each node represents a multidimensional data point, which is connected to fixed previous time points using edge with contextual information as weights 25

3.6 A depiction of time series data as minimum spanning arborescence tree 26

4.1 TODS System Overview. Reprinted with permission from [Lai et al., 2021] 28

4.2 Graphical User Interface (GUI) allow users to build and evaluate pipelines with drag-and-drop.Reprinted with permission from [Lai et al., 2021] 29

LIST OF TABLES

TABLE Page

3.1 Summary of data statistics used in our experiment. T is the number of time points, N is the number of dimensions (variables), $\sigma(\mu)$ is the standard deviation among the means of all time series (variables), and $\sigma(\sigma)$ is the standard deviation of the standard deviation of all time series. 18

3.2 Experiment results of time series forecasting and missing data imputation on three real-world high-dimensional time series datasets. The ND and the NRMSE are the normalized deviation and normalized rooted means squared error (the lower, the better). We report baseline results that are suitable for the particular dataset. 19

3.3 Anomaly detection on real-world datasets, where Pre., Rec., and F1 denote average precision, average recall, and average f1-score of all time series. 22

A.1 Experiment details of Anomaly Detection on Yahoo Dataset. 40

A.2 Continued Experiment details of Anomaly Detection on Yahoo Dataset. 41

A.3 Experiment details of Anomaly Detection on Sever Machine Dataset. 42

1. INTRODUCTION

An Anomaly is a data object that deviates significantly from the rest of the data objects in a data set. In past, the task of detecting these anomalous data points was done based on analysis by humans. With advancement in data driven approaches, detection of these anomalous data points can be done more effectively using mathematical methods. There are wide range of applications of anomaly detection tasks including detection of fraud in banking industry, detecting external attacks on servers, fault in power systems, abnormal behaviour in medical field, faults in manufacturing units etc. Usually these real world data are of form of multivariate time series and are at the centre of research of many fields. This thesis focuses of detection on anomaly on such high dimensional time series data.

Anomaly in time series data can be grouped into three main categories :

- Global Anomaly : A data point is considered a global anomaly if its value differs significantly over the aggregate of the other data points in the set.

- Contextual Anomaly : These are the points whose behaviour deviates from the rest of the data points, which are in the same context as the anomalous points. These points might be a having a normal behaviour in compare to behaviour of data points in some other context or in the entirety of all data points in the set.

- Collective Anomaly : Global and Context anomalies are a single data point. Collective anomalies are a subset of continuous points which deviates remarkably from other data points in the data.

Typically, these real-world time-series data are observed over a long period of time and so the methods dealing with these data needs to be salable for multivariate data type. These data points are also interdependent on each other. For example, the data to detect anomalous behaviour of a machine unit are usually collected using multiple sensors to capture various attributes like temperature of machine, power usage etc, in seconds across several months or in the task to find abnormality in medical field, data from various parts of the human body are monitored for weeks.

To detect these anomalous behaviour, understanding implicit characteristics of these high dimensional time series data is an important task. For which it is critical to extract temporal correlation among data points in the time series and at the same time understand inter dependence of time series among themselves, while simultaneously mitigating challenges of high dimensionality and noise in data. Finally, normal characteristics learned from these implicit feature are used to model the normal behaviour of the time series and data points whose behaviour deviates remarkably from the modelled normal behaviour of data points in the context of its neighbouring data points are considered as an anomaly. The deviation is generally measured in terms of difference between reconstructed data using the learned model and the original data.

While many end to end deep learning models uses complex neural architecture to learn temporal correlations and inter dependence, most of them are task specific deterministic models and so their conventional Mean Square Based (L2 norm) loss function puts more penalty on large reconstruction errors caused by anomalous points, which makes the learned weights biased towards these large anomalous data points. In order to overcome this, many deep learning models uses data pre-processing techniques like normalization which leads to loss of many important features.

To address this issue, The first part of this thesis explores non conventional optimization technique involving approx Mean Absolute Error(L1 norm) based loss function, which puts less penalty on large residual error points and at the same time can learn important characteristics of time series data. To observe the advantage of L1 norm based optimization function on high dimensional time series data having anomalies, Matrix factorization is an intuitive choice, as it is a simple model and is widely used to extract information from high-dimensional time series data [Yu et al., 2016].

The multi dimensional time series data can be depicted as a matrix [Figure 1.1] in which, each row of the matrix represents a single dimension of the time series and each columns represents various captured information at a particular time point. The concept of autoregression has influenced

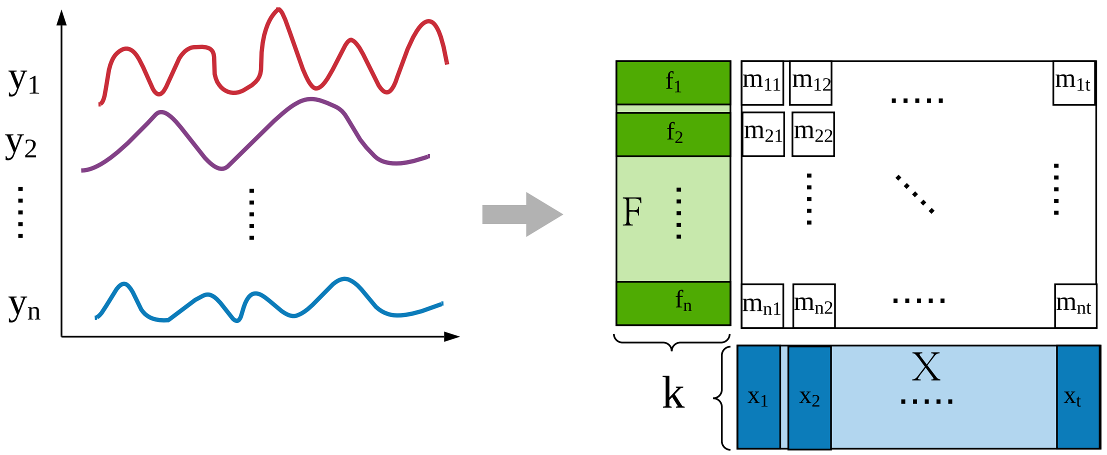


Figure 1.1: The factorization of multivariate time series matrix. A multivariate time series signal can be seen as an n by t matrix. The goal is to preserve the temporal information in k-dimensional latent features X and to model the correlation between variables in k-dimensional latent features F.

several works in this field and most of those proposed works [Chen and Cichocki, 2005, Feng et al., 2014, Zhang et al., 2009, Xiong et al., 2010, Yu et al., 2016] are focused on regularization i.e. modelling of the temporal correlation is done using various regularization terms. However, non-stationarity is a common characteristic of real world time series data, meaning simply leveraging regularization terms may cause the model to concentrate mainly on global patterns while loosing the local focus [Sen et al., 2019]. Finally, to avoid the biasness of learned weights towards anomalous data points, a differentiable anomaly-aware penalty function, based on Mean Absolute Error (MAE), [Boyd and Vandenberghe, 2004] is needed.

To address these challenges, this thesis propose the anomaly-aware temporal matrix factorization (*ATMF*). Motivated by the neigh-borhood model [Koren, 2008], a temporal neigh-borhood model is designed with two temporal bias terms: the autoregressive bias and the moving average bias. Specifically, the autoregressive bias models the temporal correlation within the temporal neighborhood of each time point; and the moving averaging bias smoothens the random noises in data and addresses the non-stationarity. To validate effectiveness of proposed bias terms , in modelling normal implicit and explicit characteristics of time series data , ATMF was tested on two classical time series analysis tasks: future data forecasting and missing data imputation. Then usage of differentiable approximation of L1 norm based loss optimization technique for matrix factorization was used for real-world time series anomaly detection task. This thesis also suggests further modification of the algorithm by encoding contextual information in the algorithm. The time series data can be compared with a directed acyclic graph where each data points can be considered as a node and contextual information like similarity based of euclidean distance can be used as edge weights. This directed graph can be used to derive minimum spanning tree for directed graph using Chu–Liu/Edmonds' algorithm and backwards walk on this can be done to chose contextual neighbours.

The second part of this thesis discusses contribution to another different challenge in time series anomaly detection task. Time series anomaly detection task has been explored extensively in scientific community. Researchers have proposed several machine learning models but yet the building a particular task specific model still prominently relies on human expertise. The selection of a good pipeline is still labour intensive and has to chose most apt data processing module, detection algorithms and also parameters for each modules. To address this challenge, Time series Outlier Detection System (*TODS*), developed by Data Lab Texas A&M aggregated and implemented commonly used time series data prepossessing, time and frequency domain based feature analyser and commonly used deterministic and stochastic outlier detection algorithms under a finite search space and provides searcher to search for best pipeline for a particular dataset using data driven techniques.

The main contributions of this thesis can be summarised as follows :

- Anomaly Aware Matrix Factorization which uses Autoregressive and moving average bias along with anomaly aware MAE loss based differentiable optimization technique to learn latent features from high dimensional time series data and use these learned feature to detect outliers.

- Implementation of known data prepossessing, feature analyser and outlier detection algorithms along with data driven searcher and graphical user interface for these modules in a already existing unified package , Time Series Outlier Detection (*TODS*).

2. RELATED WORKS

Anomaly Detection and modelling of time series data has been researched extensively in past couple of decades. In this section we discuss few related works done in this field. First, we will discuss classical approaches to time series modelling, followed by brief introduction about matrix factorization methods and extraction of latent features. Then, we will talk about time domain and frequency domain feature analysis and finally anomaly detection algorithms.

2.1 Classical Time-Series Models

One of the classical model to learn temporal correlations to analyze time series data, the traditional autoregressive model (AR)[Box and Jenkins, 1990] assumes that each time point is linearly correlated to a range of previous time points. Give a time series signal $y = \{x_1, x_2 \ldots x_t\}$, the optimization function of AR model can be formulated:

$$\min_{\phi \in R^w} \sum_{i=1}^{t} ||x_i - (ck + \sum_{j=1}^{w} \phi_j x_{i-j} + \epsilon_t)||_2,$$

In this optimization function, the second term is the predicted value, k is the predefined constant bias, ϵ is the white noise, w is the given window size indicating the number of previous time points to consider and ϕ is the parameter that models the correlation between the current time point i and previous time point j.

Real world time series data are generally noisy and non-stationary (mean of data is not constant). To address these two issues moving average is extensively used in AR modelling for dampening the effect of random noises and amplifying the long-term trend of the time series data. The moving average process basically employs a fixed length window to scan the time series and replace the last value of the window with the mean of all values inside it. The AR model was primarily introduced to model 1-d time series data. To extend its concept for multivariate time series data, various modifications has been proposed including dynamic linear model [Kalman, 1960, West and Harrison, 2006] and vector complexity [Stock and Watson, 2001]. However, due to the limitation of computational complexity [Wang et al., 2019, Yu et al., 2016], using these models for high dimensional data is not favourable. In compare with classical time series modelling methods, matrix factorization methods [Wang and Zhang, 2012] can be leveraged to model the temporal information of time series data and the correlations between concurrent features into low dimensional feature vectors.

2.2 Matrix Factorization for High-dimensional Time-series Analysis

Matrix Factorization is an extension of idea of Singular Value Decomposition(SVD) technique to factorize real or complex matrix , which itself is generalization of eigen value decomposition of a square normal matrix to any $m \times n$ matrix via an extension of the polar decomposition. MF uses optimization techniques to learn latent feature vectors from the high-dimensional data. Each entry in the input data is reconstructed by certain dimensions of latent factor vectors. The optimization function of MF based decomposition can be formulated as follows:

$$\min_{F \in R^{m \times k}, X \in R^{n \times k}} ||Y - FX^\top||_2^2 + \lambda_F ||F||_2^2 + \lambda_X ||X||_2^2,$$

where Y, is the original matrix of n by m dimension, F and X are latent features matrices, $||.||_2^2$ is the entry-wise squared Euclidean norm (L2 norm) and k is the dimension of latent factors. The first term of the objective function is the residual minimization term and the rest are the regularization terms. The residual minimization term learns the coefficients of the estimated function by minimizing the error. The regularization term with control parameter λ prevents the model from overfitting to frequent data instances by keeping the magnitude of learned parameters as small as possible.

Several works [Chen and Cichocki, 2005, Feng et al., 2014, Zhang et al., 2009, Xiong et al., 2010] tried to enhance the concept of matrix factorization by adopting graph-based regularization to model the temporal correlation for time-series data. However, since the correlations between time points in time series can be either positive or negative but the correlations (weights) be-

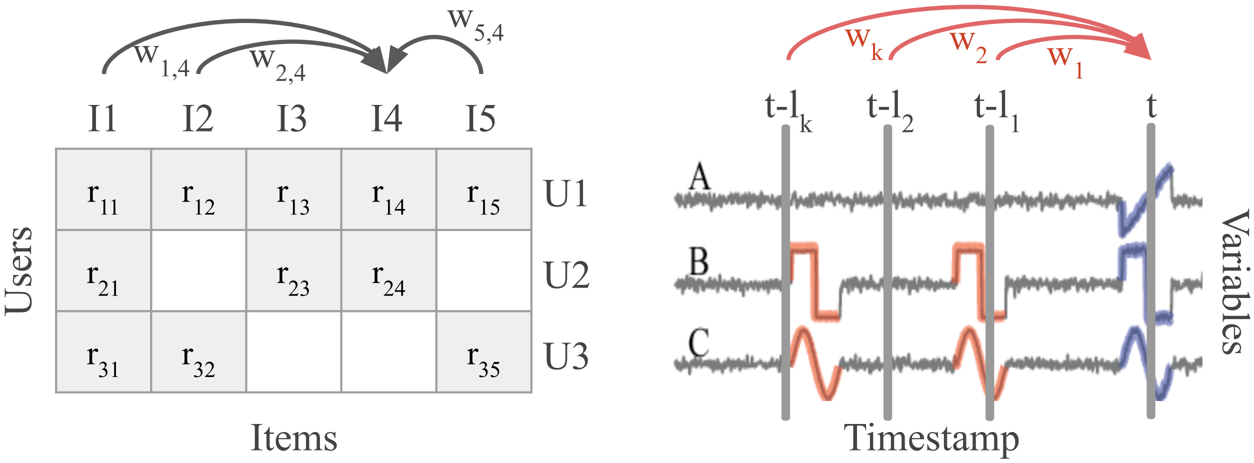


Figure 2.1: An illustration showing the similarity between neighborhood model for item-based collaborative filtering (left) and autoregressive model for time series modeling (right). As the figure shown, both models capture the correlation between the current data point and its neighbors. In the neighborhood model, the neighboring items are used to predict the missing rating of U3 on I4. The autoregressive model leverages the past neighboring points to predict the current time point.

tween vertices in graphs are positive, graph-based approaches is unsuitable to model the temporal correlation. To address the problem, temporal regularized matrix factorization [Yu et al., 2016] introduces autoregression into regularization term to model the temporal correlation within time series. However, because real-world time series signals are non-stationary and contain complex noises, directly adopting autoregression to model the temporal correlation is suboptimal. In addition, modeling temporal correlation in regularization terms makes it infeasible to incorporate statistical information such as moving average during the optimization because derivative of any constant value results in zero.

2.3 Time Domain and Frequency Domain Features Analyser

The time domain features of time series data such as mean, median, standard deviation, geometric mean, harmonic mean, energy, skewness, kurtosis, temporal derivative, willison amplitude, seasonality and trend decomposition etc can be directly used to define rules to detect abnormal behaviour in time series data.

In statistics many function estimator[Welch and Bishop, 1995] are used for estimation of joint probability distribution parameters and anomalies are separated based on their probability of occurence[Ting et al., 2007].

Many frequency domain transformation techniques like Fast Fourier Transform [Nandi and Ahmed, 2019], Discrete Cosine Transform and Wavelet transform [Narasimhan et al., 2011] are used to both extract features and also detect outlier by converting the data back into time domain and using a threshold based method to separate anomalies from normal points. One prominent work by microsoft [Ren et al., 2019] feeds saliency map features of time series data to convolution neural network to learn dynamic threshold to separate outliers from normal data points.

2.4 Anomaly Detection

The supervised learning methods, though being generally more popular in machine learning domain. In anomaly detection these methods [Park et al., 2017, Rodriguez et al., 2010] needs labeling of data with known anomaly types to train the model which could be used later to separate normal points from anomalous points. But labelling of high dimensional and large time series data set is not feasible and even merging labels of all dimensions separately can't guarantee correct labels for interdependent dimensions. Because of these limitation of supervised methods have limited usage. Another very popular method include clustering based method in which data points away from dominant cluster are marked as anomalies. Unsupervised methods for anomaly detection generally learns implicit features in low dimensional vectors and uses reconstruction error to predict anomalies. The current state of the art methods for detecting anomalies in multivariate time series can be mainly categorized into following two types:

- Deterministic Algorithms : Some of the most famous deterministic algorithms [Filonov et al., 2016, Hundman et al., 2018, Malhotra et al., 2016a] uses recurrent cells to learn from past time point experience in multivariate time series data as hidden state vector and uses it to predict the current time point. The reconstruction errors are used to separate spacecraft anomalies. To remove the curse of dimensionality [Malhotra et al., 2016a] proposes an LSTM-based EncoderDecoder to learn latent dimensions of the mutivariate data. The latent dimension are used to reconstruct the input and reconstruction error combined with statistical techniques are used to separate anomalies. These models still needs resources to store all the learned parameters.

- Stochastic Algorithms : Another popular set of algorithms uses stochastic optimization techniques [Park et al., 2017, Zong et al., 2018] to model time series data. These works have suggested that use of stochastic variable in Recurrent Neural Networks (RNN) can help to model uncertainty in time series data and thus improves the overall performance. [Park et al., 2017] proposes a model DAGMM which joins Deep Autoencoder (AE) and Gaussian Mixture Model (GMM) simultaneously. AE is used to reduce the dimension of data and GMM is used to estimate the probabilistic density of representation. Although, this method is designed for multivariate variables, it doesn't achieved success on time series data as it doesn't models the temporal correlation among time points. Another very popular model [Park et al., 2017] learns temporal correlation by LSTM and overcomes dimesionality curse by using VAE. It uses LSTM as the feedforward layer of VAE.

3. ANOMALY AWARE MATRIX FACTORIZATION

A time series contains successive observations which are usually collected at equal-space timestamps. In our study, we focus on multi dimension time series, defined as $x = x_1, x_2, ..., x_N$, where N is the length of x, and an observation $x_t \in R^M$ is an M-dimensional vector at time $t(t \leq N) : x_t = [x_t^1, x_t^2, ..., x_t^M]$ and $x \in R^{M \times N}$. For multivariate time series anomaly detection, the objective is to determine whether an observation x_t is anomalous or not.

To use matrix factorization for anomaly detection task, first depiction of the multivariate time series as a matrix is needed. Then, the main objective of the problem, to learn latent features using decomposition of this matrix into matrices of smaller dimensions can be formulated as an optimization problem. The learned latent features are then used to reconstruct the original matrix and the reconstruction error is used for detecting anomalies.

Let $y = \{m_1, m_2 \dots m_t\}$ be a fully observed time series, where $m_t \in R$ is the signal observed at the time point t. A multivariate time series can be defined as $\mathbf{Y} = \{y_1, y_2, \dots, y_{n-1}, y_n\}$, where n is the variable dimension. This multivariate time series $\mathbf{Y} \in R^{n \times t}$ can be formulated as an n by t matrix. Now, given a multivariate time series matrix $\mathbf{Y}$, the problem of latent feature extraction is formulated as:

$$\min_{F \in R^{n \times k}, X \in R^{t \times k}} \mathcal{L}(\mathbf{Y}, \mathcal{P}(F, X^\top))$$

where $\mathcal{P}$ is the predictor function which aims to reconstruct the input matrix $\mathbf{Y}$ by updating the k-dimensional latent features F and X, and $\mathcal{L}$ is the loss function which estimates the error between the input matrix and the predicted matrix and penalizes the instances with estimation errors. Note that, F is the latent feature for each variable and X is the latent feature for each time point. As the figure 1.1 shows, the goal of our problem is to map the given multivariate time series matrix into the low-dimensional vector space and also to preserve the temporal correlations between the time points and the inter dependency between the variables.

3.1 Methodology

In this section, first the temporal neighborhood model is introduced and then the two temporal biases for matrix factorization are illustrated. After this, further investigation of the loss function of the residual minimization term and derivation of an anomaly-aware loss function are done with theoretical analysis. Finally, the temporal neighborhood model is integrated with an anomaly-aware penalty function and an anomaly-aware temporal-biased matrix factorization (ATMF) model is proposed.

3.1.1 Temporal Neighborhood Model

Collaborative filtering based recommendation systems widely uses Neighborhood models [Koren, 2008]. Under the intuition that similar users tend to have similar biases towards items, [Bell and Koren, 2007] introduces the biases of users with interpolation weights to model the similarity and correlation weights between users and items. The model predicts the ratings of unseen items for each user by linearly combining the rating of observed similar items using the learned weights.

The similar items in item recommendation system can be compared with temporal neighbours in time series data. Taking this comparison as an insight, in this thesis, temporal neighborhood model was designed. This temporal neighborhood aims to model the implicit temporal information among the given data points. The designed neighborhood model specifically, has two main bias terms for modelling different aspects of data. The first temporal bias term: the autoregressive bias aims to model correlations within the temporal neighborhood and the moving average address the non-stationary property of time series data and helps in smoothing the noise in data.

3.1.2 Autoregressive Bias

Taking the neighborhood model into consideration, autoregression defined earlier in section 2.1 , can be seen as one of its variants. In AR each time point is a linear combination of the preceding time points. These preceding time points can be considered as similar neighbours. That is, the aim should be to learn the correlation between the past time points with the current time points, so that the prediction of the current time point is as precise as possible. This similarity between the neighborhood model and the autoregression model is illustrated in figure 2.1 . Based on the notion of autoregression, the **autoregressive bias** is defined for time point t as follows:

$$B_t^{AR} = \sum_{l \in L} W_l^\top \mathbf{Y}_{t-l}$$

where $\mathbf{Y}_t \in R^{n \times 1}$ is the t-th column vector of the time series matrix $\mathbf{Y}$, n is the number of time series, L is the lag set, and the vector W_l from $W \in R^{n \times L}$ models the temporal correlations between the t-th and $(t-l)$-th time points.

3.1.3 Moving Average Bias

Typically, **moving average** (MA) is incorporated into autoregressive models to address the non-stationary and random noise problems in time series data. However, this thesis uses MA as a statistical value and derivative of constant value is zero. So if kept inside the regularization term of objective function, whose main objective is to keep the absolute vales learned weights small, MA will disappear in an derivative based optimization technique. To overcome this limitation, MA is formulated as a bias term:

$$B_t^{MA} = \frac{\sum_{l=0}^{w} \mathbf{Y}_{t-l}}{w},$$

where w is a hyper-parameter indicating the window size. Thus, the moving average values can be preserved in the residual term. Note that, different from the typical moving average process, which considers both the previous time points and the future time points when calculating the current time point, only the previous time points are used in this formulatoin when calculating the mean value.

3.1.4 Temporal Biased Matrix Factorization

After merging the two temporal bias terms defined above in the residual part of objective function, the predictor function $\mathcal{P}$ is defined as follows:

$$\mathcal{P}(F, X^\top, W) = FX^\top + B^{AR} + B^{MA},$$

where the first term leverages the learned latent features, and the second and the third terms the temporal correlation within the temporal neighborhood .

Based on the predictor function, the objective function is then derived as a mean square error penalty function with regularization terms for each parameter:

$$\min_{F \in R^{n \times k}, X \in R^{t \times k}, W \in R^{n \times L}} ||\mathbf{Y} - \mathcal{P}(F, X^\top, W)||_2^2 + \lambda(||X||_2^2 + ||F||_2^2 + ||W||_2^2) \tag{3.1}$$

To optimize Equation 3.1, the batch gradient descent is employed in this formulation to accelerate the training process. Specifically, in each training iteration, the gradient of all training data is first calculated and then all of the parameters are updated at once after receiving all of the gradients. For convenience, E is used to denote the residual term $\mathbf{Y} - \mathcal{P}(F, X^\top)$, γ is used as learning rate, and λ as the regularization control parameter. The parameters are updated by moving in the opposite direction of the gradient. The update rules for all learned parameters are as follows:

$$F \leftarrow F + \gamma(EX - 2\lambda F)$$

$$X \leftarrow X + \gamma(E^\top F - 2\lambda X)$$

$$W_l \leftarrow W_l + \gamma(E_t \circ \mathbf{Y}_{t-l} - 2\lambda W_l)$$

, where $\circ$ is the matrix multiplication operator.

3.1.5 Anomaly-aware Penalty Function

The mean square error (MSE)(L2 norm) based penalty function employed by traditional matrix factorization has some inherent limitations to it. To overcome these limitation, an alternative anomaly-aware penalty function based on mean absolute error (MAE)(L1 norm) is used in our formulation. Although incorporating moving average bias can smooth out the anomalies with low deviation, those anomalies with high deviation will still affect the trend of the time series and make the prediction more challenging. Ideally, if extreme anomalies can be identified during the learning process, then their effects can dampened by lowering their learned weights. However, the MSE penalty function is sensitive towards anomalies as they have large residuals and thus in order to reduce the residuals, MSE actually exacerbates anomalies detection task by giving more weightage to anomalous points while learning, effecting other normal data points. Instead, use of MAE based optimzation is favourable in for AD task.

Figure 3.1 illustrates the different penalizing effects of mean square error (MSE) and mean absolute error (MAE). Specifically, MSE (squared L2 norm, i.e. x^2) has quadratic growth and thus puts more penalty on the large residuals caused by anomalies, which biases the model to fit the anomalies instead of normal data. On the other hand, MAE (L1 norm, i.e. $|x|$) grows linearly and thus penalizes residuals in proportion to their magnitudes. Therefore, MAE is able to penalize large residuals without succumbing to the pitfall of anomalies, making MAE an excellent anomaly-aware penalty function when handling real-world data.

One big problem with , MAE is it's non differentiablity at 0 and thus is incompatible with differentiation based convex optimization methods. To address the problem, a differentiable penalty function based on MAE is used which estimates $|x|$ with $\sqrt{x^2 + \epsilon}$, where ϵ is a negligibly small positive real number, and x represents the residual between the ground truth and the predicted value. To ensure the convergence of the objective (equation 3.2) with the proposed penalty function, this thesis first shows the differentiability and convexity of the proposed penalty function : Let $\mathcal{H}(x) = \sqrt{(x+h)^2 + \epsilon}$. $\forall x \in R$, $\mathcal{H}(x)$ is a differentiable convex function.

Proof. To show the $\mathcal{H}(x)$ is differentiable, we first show that, $\forall x \in R$, the derivative $\mathcal{H}'(x)$ exists:

$$\begin{aligned}
\mathcal{H}'(x) &= \lim_{h\to 0} \frac{(\sqrt{(x+h)^2+\epsilon} - \sqrt{x^2+\epsilon})}{h} \\
&= \lim_{h\to 0} \frac{(\sqrt{(x+h)^2+\epsilon} - \sqrt{x^2+\epsilon})}{h} \\
&\quad \times \frac{(\sqrt{(x+h)^2+\epsilon} + \sqrt{x^2+\epsilon})}{(\sqrt{(x+h)^2+\epsilon} + \sqrt{x^2+\epsilon})} \\
&= \lim_{h\to 0} \frac{((x+h)^2+\epsilon) - (x^2+\epsilon)}{h(\sqrt{(x+h)^2+\epsilon} + \sqrt{x^2+\epsilon})} \\
&= \lim_{h\to 0} \frac{h^2+2xh}{h(\sqrt{(x+h)^2+\epsilon} + \sqrt{x^2+\epsilon})} \\
&= \frac{x}{\sqrt{x^2+\epsilon}}
\end{aligned}$$

since, $\frac{x}{\sqrt{x^2+\epsilon}}$ is finite for all $x \in R$, $\mathcal{H}(x)$ is differentiable. Now, the second derivative of $\mathcal{H}(x)$ i.e. $\mathcal{H}''(x) = \frac{\epsilon}{(1+x^2)^{\frac{3}{2}}}$ is greater than 0 for all $x \in R$, so based on the second order derivative condition of convexity, the proposed penalty function is convex also. □

3.2 Anomaly-aware Temporal Matrix Factorization

With the proposed anomaly-aware penalty function, the L2 norm based objective function proposed in equation 3.1 can be modified to incorporate L1 norm benefits as follows:

$$\min_{F\in R^{n\times k}, X\in R^{t\times k}, W\in R^{n\times L}} \sqrt{||\mathbf{Y} - \mathcal{P}(F, X^\top, W)||_2^2 + \epsilon} \quad + \lambda(||X||_2^2 + ||F||_2^2 + ||W||_2^2) \tag{3.2}$$

To show that global optimum for objective 3.2 exists for any gradient solver, it is needed show that that objective function 3.2 is convex. The residual part of the objective function can be written as $\mathcal{H}(E)$, where $E = \mathbf{Y} - \mathcal{P}(F, X^\top, W)$ is the residual matrix, and $\mathcal{H}(x)$ denotes the penalty function defined above . Based on the composition rule of convexity [Boyd and Vandenberghe, 2004], the equation 3.2 is convex if both $\mathcal{H}(x)$ and $||E||_2^2$ are convex and $\mathcal{H}(x)$ is non-decreasing,

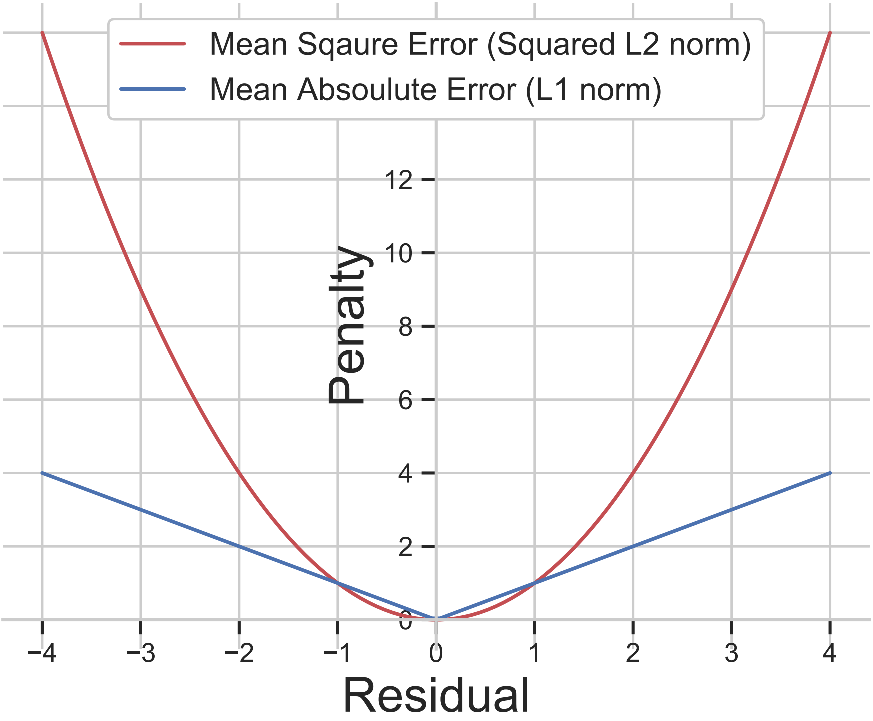


Figure 3.1: As residual grows, the MSE (square L2 norm) and MAE (L1 norm) assign the penalties with quadratic growth and linear growth, respectively.

The same condition also holds for vector functions. Thus, to prove the convexity of $||E||_2^2$, we follow the proof that all norm functions are convex [Boyd and Vandenberghe, 2004] if the argument is piece-wise linear function and derive the following lemma: If $\mathcal{G}(E) = ||E||_2^2$ and E is piece wise linear function, then $\mathcal{G}(E)$ is a convex function. Note that, all of the regularization terms are norm functions and therefore are convex. In addition, the residual term of objective 3.2 can be represented as $\mathcal{H}(\mathcal{G}(E))$, where the domain of $\mathcal{H}(.)$ equals to the range of $\mathcal{G}(E) = ||E||_2^2$. Since the first derivative $\mathcal{H}(x)$ i.e.$\frac{x}{\sqrt{x^2+\epsilon}} \geq 0$ when $x \geq 0$ and $\mathcal{G}(x) = ||E||_2^2$ is always greater than or equal to 0, $\mathcal{H}(x)$ is a non-decreasing function. Based on lemma 3.1.5 and 3.2 and our previous proof, both the residual and the regularization terms are convex. Thus, the proposed objective function is convex and any gradient solver can reach the global optimum.

Let l be the learning rate and λ be the regularization control parameter. To solve equation 3.2 with batch gradient descent, the following update rules are derived:

$$F \leftarrow F + l((E \oslash \sqrt{E \circ E + \epsilon J})X - 2\lambda X)$$

$$X \leftarrow X + l((E \oslash \sqrt{E \circ E + \epsilon J})^\top F - 2\lambda X)$$

$$W_l \leftarrow W_l + l((E_t \oslash \sqrt{E_t \circ E_t + \epsilon J_t}) \circ \mathbf{Y}_{t-l} - 2\lambda W_l)$$

where $\sqrt{\cdot}$, $\circ$, and $\oslash$ are element-wise square root, multiplication, and division functions for matrix, respectively, and J is a matrix of ones.

3.3 Experiment and Analysis

In this section, the proposed anomaly-aware matrix factorization algorithm is evaluated. The experiments were designed to answer the following research questions:

- How is *ATMF* compared against the existing methods in modelling features of high-dimensional time series analysis using two classical tasks, i.e., data imputation and time series forecasting?
- Does the proposed anomaly-aware residual minimization term enable *ATMF* to detect anomalies form real-world non-stationary time series data?
- How does the proposed MAE loss function detect anomalies? How do approx MAE based losses react differently compared to MSE losses towards anomalies?

3.3.1 Experiment Settings and Datasets

To demonstrate the effectiveness of two bias terms in modelling the time series, the ATMF was evaluated on two classical time series analysis tasks, i.e., time series forecasting and missing data imputation on 3 real world dataset and then was evaluated on the anomaly detection task using 2 publicly available real-world datasets. The dataset statistics are tabulated in Table 3.1. More experiment details can be found in the Appendix.

3.3.1.1 Time Series Analysis

Three datasets were considered for the time series analysis tasks: traffic, electricity, and

Dataset	T	N	$\sigma(\mu)$	$\sigma(\sigma)$
Traffic	25,968	370	1.19e+4	7.99e+3
Electricity	10,392	963	1.08e-2	1.25e-2
Wikipedia	747	115,084	4.85e+4	1.26e+4
Yahoo-A1	1,400	1	7.15e+5	1.29e+5
SMD	28,479	38	1.87e-1	6.81e-2

Table 3.1: Summary of data statistics used in our experiment. T is the number of time points, N is the number of dimensions (variables), $\sigma(\mu)$ is the standard deviation among the means of all time series (variables), and $\sigma(\sigma)$ is the standard deviation of the standard deviation of all time series.

wikipedia. The Traffic dataset is a collection of 15 months' worth of daily data from the California Department of Transportation. The data describes the occupancy rate (between 0 and 1) of different car lanes of San Francisco bay area freeways. The Electricity dataset records the electric power consumption of 370 households for 47 months. The Wikipedia dataset contains the web traffic of 115,084 Wikipedia articles. And the time series represent the daily view counts of different articles from July 2015 to December 2016.

For all of the datasets, following the previous works [Yu et al., 2016, Sen et al., 2019], data are first z-normalized. For the time series forecasting task, the last 24 time points for each variable is predicted based on the fully observed historical data. For the missing data imputation task, 40% of the data entries are randomly masked and their values are imputed. Normalized deviation (ND) and normalized rooted mean square error (NRMSE) are adopted as evaluation metrics following the previous works [Yu et al., 2016, Sen et al., 2019].

3.3.1.2 Anomaly Detection

Two large-scale real-world datasets is considered for time series anomaly detection: Yahoo [1] and Server Machine Dataset [2] (SMD), for detecting anomalies from univariate and multivariate time series data, respectively. The Yahoo dataset contains the web traffic data with labels that are editorially generated by the publisher. There are 67 uni-variate time series, which represent 67 Yahoo web pages. Anomaly detection experiment was performed on each time series and the

[1] https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70

[2] https://github.com/NetManAIOps/OmniAnomaly/

Dataset	Traffic (ND/NRMSE)		Electricity (ND/NRMSE)		Wikipedia (ND/NRMSE)	
1-7 Tasks	Forecasting	Imputation	Forecasting	Imputation	Forecasting	Imputation
AR(1)	0.387/1.006	-/-	0.461/0.688	-/-	194/110350	-/-
Mean	1.000/1.534	1.000/1.406	1.000/1.292	1.000/1.206	1.011/2.288	1.007/2.329
FFill	1.288/1.912	1.014/1.596	0.931/1.226	0.794/1.191	0.953/2.613	0.908/2.715
RNN-LSTM	0.509/1.066	-/ -	0.590/0.854	-/-	2.287/13.682	-/-
AE	-/-	1.001/1.403	-/-	1.000/1.206	-/-	1.003/2.320
SVD-AR(1)	0.359/**0.946**	-/-	0.484/0.716	-/-	0.850/2.167	-/-
TRMF	**0.359**/0.952	0.523/0.911	0.478/0.705	0.629/0.853	0.866/2.229	1.023/2.342
ATMF	0.383/0.980	**0.522/0.851**	**0.453/0.660**	**0.594/0.763**	**0.798/2.158**	**0.873/2.160**

Table 3.2: Experiment results of time series forecasting and missing data imputation on three real-world high-dimensional time series datasets. The ND and the NRMSE are the normalized deviation and normalized rooted means squared error (the lower, the better). We report baseline results that are suitable for the particular dataset.

average performance was reported. The SMD is a 5-week-long dataset with 23 multivariate time series representing 23 machines. In SMD anomaly detection experiment was performed on each machine and the average result was reported. Note the anomaly ratio for the Yahoo dataset and SMD are 4.16% and 1.76%, respectively. The contamination ratio was set to 0.001 for the Yahoo-A1 dataset and to 0.05 for SMD dataset following the anomaly precentage. The models were evaluated in terms of average precision, recall, and f1-score.

3.3.1.3 Baselines

Seven baselines were considered for the two classical time series analysis tasks and four baselines were considered for time series anomaly detection task. The baselines are categorized into 3 approaches: traditional, matrix factorization based, and deep neural network based.

Traditional methods :

For forecasting task, the baselines include an N-dimensional AR model (*AR(1)*) which learns time point correlations of each variable independently as the baseline for forecasting task; a statistical method that estimates the missing values by using the mean of data (*Mean*); and a forward filling algorithm (*FFill*) which fill the missing/incoming values with the last non-missing value in the time series. For anomaly detection, baselines include one-class support vector machine (*OCSVM*) [Schölkopf et al., 2000] and isolation forest (*IForest*) [Liu et al., 2008]. Each of the time point was treated as a data instance and values of the variables at the time point as the features, to detect the anomaly time points for the two traditional methods.

Matrix factorization :

In forecasting task, singular value decomposition with autoregression (*SVD-AR(1)*) first obtains singular vectors U, V and singular value D using SVD; sets the latent feature of variables F as $U \cdot D$ and latent feature of time points X as $V^\top$; and then leverages AR(1) on k-dimensional X to predict future X and to perform forecasting using the dot product of F and the predicted X. On the other hand, temporal regularized matrix factorization (*TRMF*) [Yu et al., 2016] learns the latent factor with autoregressive regularization and is baseline in all of the tasks.

Deep Neural Networks :

Recurrent neural network with LSTM units (*RNN-LSTM*) [Sak et al., 2014] is a widely used neural architecture for time series forecasting task. The autoencoder (*AE*) with fully connected layer [Beaulieu-Jones and Moore, 2017] is a well-studied neural architecture for data imputation task. For anomaly detection task, the LSTM-based encoder-decoder (*LSTM-ED*) [Malhotra et al., 2016b] was included, which leverages reconstruction error to detect anomalies.

3.3.2 Classical Time Series Tasks

To answer the first research question, the proposed model was evaluated on the two classical time series tasks: time series forecasting and missing data imputation. Table A.3 tabulates the experimental results of the two tasks on the three benchmark datasets. It can be seen, clearly *ATMF* outperforms most of the baselines in time series forecasting task with the average improvement of 1.56% over the second-best algorithm in terms of ND. In addition, *ATMF* outperforms all of the baselines in imputation task with the average improvement of 5.6% over the second-best algorithm in terms of NRMSE.

From Table A.3, it can seen that the matrix factorization based approaches is better at modeling high-dimensional time series data than the traditional methods do. Particularly, for the Wikipedia dataset, the performance of both *AR(1)* and *RNN-LSTM* are significantly worse than other methods while matrix factorization based methods maintain good performances. This is likely because while *AR (1)* and *RNN-LSTM* forecast the future points by considering only the temporal correlation in the same dimension, matrix factorization based methods do so by considering the temporal correlation of all dimensions. As for imputation task, both *TRMF* and *ATMF* significantly outperforms the *AE*. The above observations show modeling high dimensional data is crucial to the performance of both time series forecasting and missing data imputation, and matrix factorization approaches are suitable for the tasks.

Next, it was observed that using the moving average bias along with the anomaly-aware loss function significantly improves the performance of both forecasting and imputation tasks. It should be marked that the two autoregression based methods, *SVD-AR(1)* and *TRMF*, have similar performances. In contrast, as an autoregression method which incorporates the moving average bias and the anomaly-aware loss function, *ATMF* attains the best performance. In time series forecasting task on the Traffic dataset, it was noticed that *ATMF* is slightly inferior to the other two matrix factorization methods. A possible explanation is that, the smoothing effect of moving average, which is tailored for non-stationary data, can oversmooth the relatively stationary Traffic dataset and thus sacrifice some information. In contrast, *ATMF* demonstrates superior performance in the non-stationary Wikipedia dataset and Electricity dataset. As real-world data are usually non-stationary, leveraging the moving average bias and the anomaly-aware loss function of *ATMF* greatly benefits applications involving real-world time series data.

3.3.3 Anomaly Detection

To answer the second research question, experiments were conducted for both univariate and multi-variate time series anomaly detection tasks on the Yahoo dataset and SMD, respectively. To conduct anomaly detection, the data points with the highest prediction errors are identified as the anomalies for *LSTM-ED* and matrix factorization-based methods. For the two traditional methods, *OCSVM* identify the anomalies which are far from the hyperplane, and *IForest* identify the data points with shortest path length to the root node as the anomalies. Table 3.3 tabulates the

Dataset	Yahoo-A1 (Pre./Rec./F1)	SMD (Pre./Rec./F1)
IForest	0.687/0.177/0.239	0.238/0.443/0.263
OCSVM	0.687/0.176/0.240	0.236/0.471/0.276
LSTM-ED	0.678/0.186/0.253	0.215/0.443/0.253
TRMF	0.198/0.059/0.072	0.254/0.467/0.284
ATMF	**0.766/0.187/0.259**	**0.294/0.551/0.334**

Table 3.3: Anomaly detection on real-world datasets, where Pre., Rec., and F1 denote average precision, average recall, and average f1-score of all time series.

results of univariate and multivariate time series anomaly detection tested on 67 and 23 time series, respectively. As shown in the table, in terms of precision, recall, and f1-score, *ATMF* outperforms the second-best baseline by 11.4%, 0.5%, 2.3% in the univariate setting and by 15.7%, 16.9%, 17.6% in the multi-variate setting.

Based on the experiment, two observations were made. First, matrix factorization based methods are suitable for multivariate time series anomaly detection. As it can be seen, both *ATMF* and *TRMF* perform well in the multivariate time series data SMD. Second, it was observed that anomaly-aware loss function significantly improves the performance of anomaly detection. Specifically, using the traditional MSE loss function, *TRMF* can be swayed to fit the outliers and therefore produces suboptimal performance for the anomaly detection task. On the other hand, using both anomaly-aware loss function and the moving-average bias, *ATMF* can penalize outliers in proportion to the magnitudes of their errors, which allows the method to be sensitive to the change in regular data while remaining resilient to the extreme effects of anomalies. Hence, *ATMF* is able to achieve superior performance in both univariate and multivariate anomaly detection tasks trained on non-stationary real-world datasets.

3.4 Anomaly-aware Temporal Matrix Factorization

3.4.1 Case studies on MAE loss and MSE loss for anomaly detection

To answer the third research question, *ATMF* was trained with both MAE and MSE loss functions on the Yahoo dataset and their differences was studied after both converge to the same final

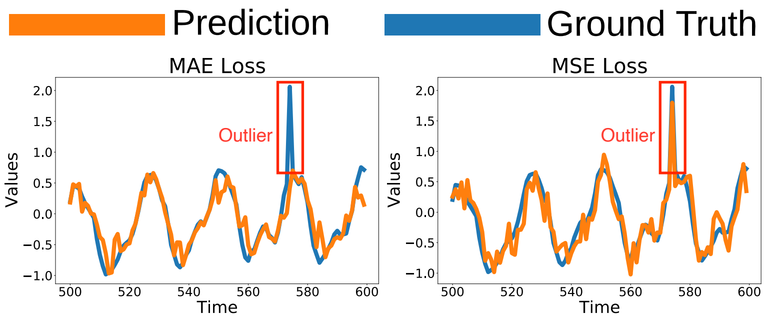


Figure 3.2: The smooth time series of Yahoo-A1 real_3

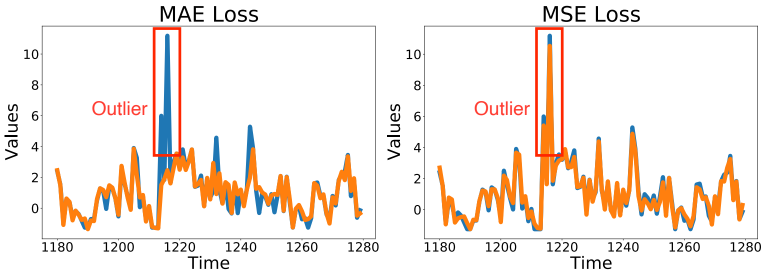


Figure 3.3: The noisy time series of Yahoo-A1 real_1

Figure 3.4: Case studies of MAE and MSE loss functions on noisy data. The MAE loss gives higher prediction error than the MSE loss does, making it more suitable for anomaly detection.

loss. Observing some time series in the Yahoo dataset are less noisy (with more regular periods) than the others, both kinds of time series was considered in this section.

Based on figure 3.1, it can be observed that using the MSE loss results in minimal error differences between the outliers and the regular data points, leading to challenging anomaly detection. This is due to the MSE loss over emphasize on penalizing the large residuals, i.e., using a quadratic function, biasing the model to fit the most anomalous data points. On the other hand, the MAE loss penalizes residuals in proportion to their magnitudes, i.e., using a linear function, which allows the model to still focus on fitting the data with large errors while guarding against extreme errors produced by outliers. Therefore, using the MAE loss leads to larger prediction errors than using the MSE loss does, making MAE ideal for anomaly detection.

3.5 Future Works

In this section, the thesis talks about the possible extension of *ATMF*, by encoding contextual information in matrix factorization. Time series data can be compared with a directed acyclic graph (DAG) ($G = (V, E)$), where each data point is depicted as a node (V) and the contextual information can be used as edge (E) weights connecting current data points with past data points. This contextual information could be of many different types includes similarity scores using any distance metrics, discordance score using matrix profile [Yeh et al., 2016] etc.

To restrict the size of DAG, each present time points can be connected to a fixed number consecutive past neighbours which is defined by a fixed size window. This contextual DAG is then changed into a spanning arborescence of minimum weight using Chu–Liu/Edmond's algorithm , which is an analog of minimum spanning tree for undirected graph. The acyclic tree $T = (V, E')$ data structure obtained by applying Chu–Liu/Edmonds will consists of all vertices i.e. data points but only a subset of edges $E' \subset E$, with minimum total sum of edge weights possible.

Figure 3.5 summarizes this procedure of depicting time seris data as DAG $G = (V, E)$ and the conversion of it into spanning arborescence tree $T = (V, E')$ of minimum weights will create a tree similar to figure 3.6. Considering a node say 7 in the figure 3.6 , its contextual neighbours by walking 3 steps backward in time will be 4 ,1 and 0 .

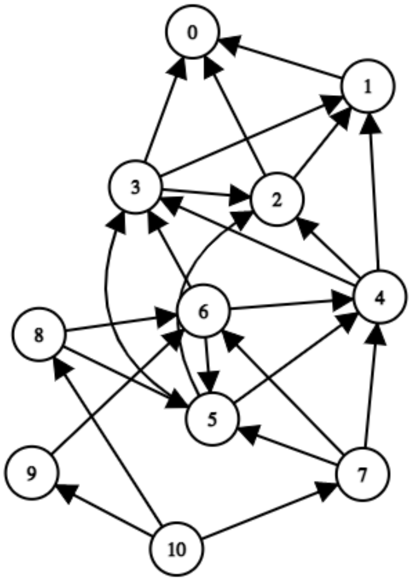


Figure 3.5: A depiction of time series data as directed acyclic graph (DAG). Each node represents a multidimensional data point, which is connected to fixed previous time points using edge with contextual information as weights

The fixed temporal neighbour used for modelling can be augmented or replaced by neighbours which are obtained by walking fixed few steps backward in the spanning arborescence tree. This can lead to two types of models , one replacing fixed neighbours in AR bias with contextual neighbours or to incorporating contextual information as a statistical value in moving average bias.

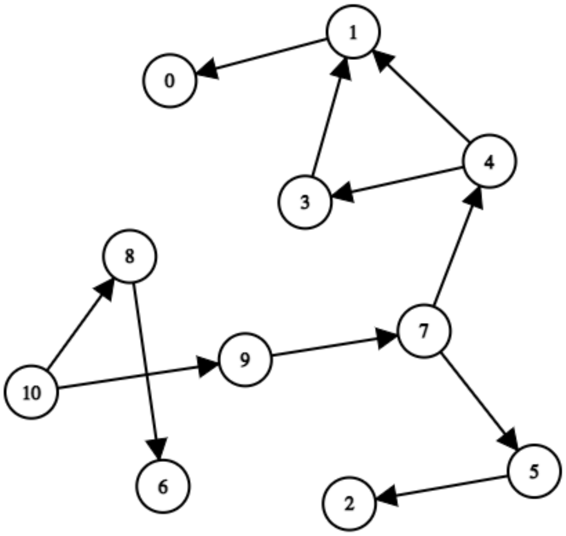


Figure 3.6: A depiction of time series data as minimum spanning arborescence tree

4. TODS: TIME SERIES OUTLIER DETECTION SYSTEM -AUGMENTATION [1]

In this part of thesis work done on an unified interface Time-series Outlier Detection System *(TODS)* is discussed. TODS is a full-stack automated machine learning system for outlier detection on multivariate time-series data develpoed by Data Lab , Texas AM University . Figure 4.1 shows overview of it's system .It currently supports 70 primitives for data processing, time series processing, feature analysis, outlier detection, and incorporating human knowledge. It can be applied to various application scenarios, including point-wise, pattern-wise and system-wise detection. The goal of TODS is providing an end-to-end solution for real-world time series outlier detection. A pipeline in TODS is defined as a directed acyclic graph, where each step represents a primitive. A typical pipeline has four primitive steps, including data processing, time series processing, feature analysis, and detection algorithms. This thesis auguments TODS by implementing some commonly used algorithms into primitives format which can be used by data driven searchers to search for best primitive to construct an effective pipeline for outlier detection on a given data set. The implemented algorithms can be categorised as follows :

- **Data Prerprocesssors :** This groups consists of primitives to process raw data, which includes conversion of dataset in dataframe format suitable for data driven searcher, parsing columns of dataset to form metadata used by searcher and time stamp continuity validation checks.

- **Time Series Preprocessors :** This group consists of a primitive [Ronald Eastman et al., 2009] for decomposition of time series into seasonality and trend which is further used by other primitives .

- **Feature Analyser :** This group consist of statistics based time domain feature analyser

[1] Reprinted with permission from "Lai, K.-H., Zha, D., Wang, G., Xu, J., Zhao, Y., Kumar, D., Chen, Y., Zumkhawaka, P., Wan, M., Martinez, D., Hu, X. (2021). TODS: An Automated Time Series Outlier Detection System". Proceedings of the AAAI Conference on Artificial Intelligence, 35(18), 16060-16062.Copyright © 2021, Association for the Advancement of Artificial Intelligence

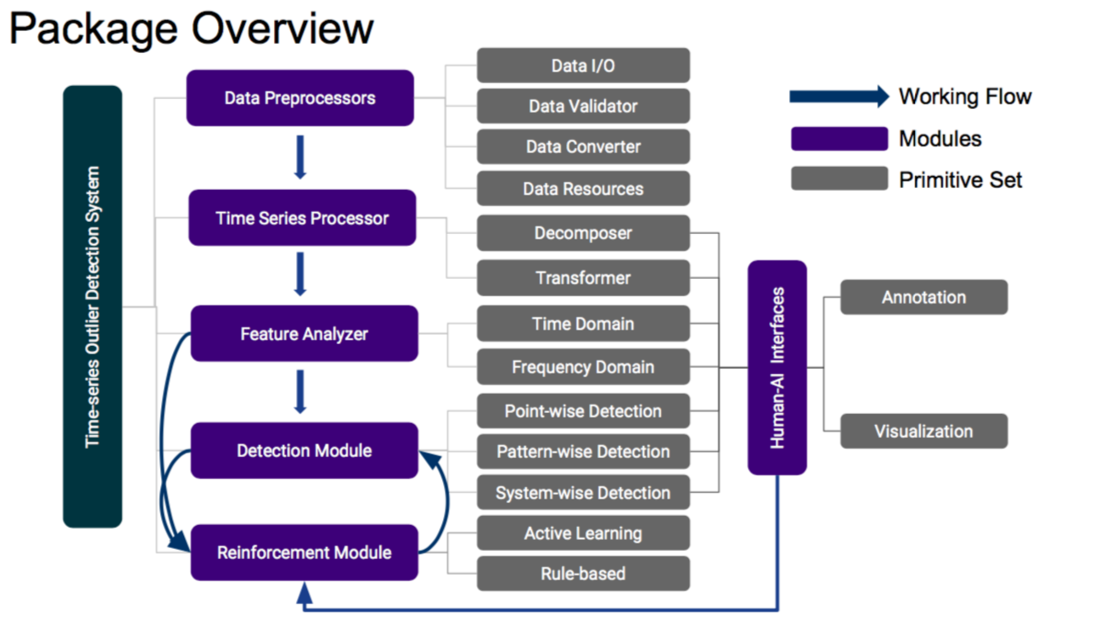
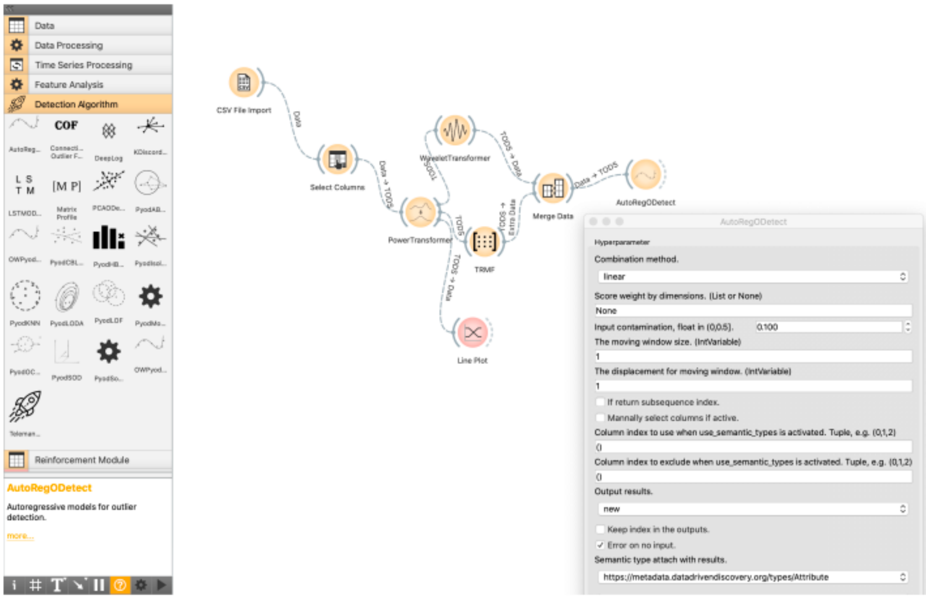


Figure 4.1: TODS System Overview. Reprinted with permission from [Lai et al., 2021]

and spectral transform based frequency domain feature analyser . Statistical feature Analyser includes primitives for Mean, Median, Standard deviation, Variation, Variance, Vector Sum, Kurtosis, Skewness , Absolute Energy, Geometric Mean, Harmonic Mean, Mean and Absolute Mean Temporal Derivative, Willison Amplitude, Median Absolute Deviation and Zero Crossing. Frequency Domain feature analyser includes primtive for Spectral Residual Transform , which converts time domain data to spectral domain using Fourier transform and converts back to time domain using Inverse Fourier transform .

- **Detection Algorithms :** This groups consists of already known Deterministic and Stochastic algorithms. Deterministic algorithms based primitives includes Long short-term memory based "DeepLog" [Du et al., 2017] and Stochastic algorithms based primitive includes Generative Adversarial Active Learning for Unsupervised Outlier Detection [Liu et al., 2018] which employs generative adversarial learning to directly generate informative potential outliers, solving the lack of information caused by the ”curse of dimensionality” and Deep AutoEncoding Gaussian Mixture Model [Zong et al., 2018]. which jointly optimizes the



Autoregressive models for outlier detection.

Hyperparameter

Combination method.

linear

Score weight by dimensions. (List or None)

None

Input contamination, float in (0,0.5]. 0.100

The moving window size. (IntVariable)

1

The displacement for moving window. (IntVariable)

1

□ If return subsequence index.

□ Mannally select columns if active.

Column index to use when use_semantic_types is activated. Tuple, e.g. (0,1,2)

()

Column index to exclude when use_semantic_types is activated. Tuple, e.g. (0,1,2)

()

Output results.

new

□ Keep index in the outputs.

☑ Error on no input.

Semantic type attach with results.

https://metadata.datadrivendiscovery.org/types/Attribute

Figure 4.2: Graphical User Interface (GUI) allow users to build and evaluate pipelines with drag-and-drop.Reprinted with permission from [Lai et al., 2021]

parameters of the deep autoencoder and the mixture model simultaneously in an end-to-end fashion.

In addition to this, the graphical user interface (GUI) for each primitive allows users to build and evaluate pipelines with drag-and-drop and also incorporate human in loop through support for visualization.

5. CONCLUSION

In this thesis, one matrix factorization based algorithm *ATMF* and a package *TODS* for automated time series outlier detection is discussed. *ATMF*, is proposed as an anomaly-aware matrix factorization method to model the high-dimensional and non-stationary time series data. Specifically, a temporal neighborhood model was proposed with two temporal biases to address the temporal dependent and non-stationary attributes of time series data. Furthermore, we design an anomaly-aware penalty function to tackle the outliers within the real-world data and proved its convexity. The results in three application scenarios demonstrate that the *ATMF* outperforms all of the baselines ranging from the traditional time series models, deep learning based models, and other matrix factorization based models. We further suggested extension of this work by encoding any type of contextual information into model. To encode the contextual information the time series data was depicted as a DAG and was converted to spanning arborescence of minimum weights using Chu–Liu/Edmonds' algorithm. Fixed steps walk on this spanning tree could be used to generate contextual neighbourhood matrix.

At last this thesis briefly mentions *TODS* which supports 70 primitives for data processing, time series processing, feature analysis, outlier detection, and incorporating human knowledge. It also provides data driven searcher and a human in loop functionality.

5.1 Further Studies

The models proposed in this paper are being improved continuously by enhancing its learning ability and computation efficiency. *TODS*, being an open end project many more new algorithms can be implemented as primitives .

REFERENCES

B. K. Beaulieu-Jones and J. H. Moore. Missing data imputation in the electronic health record using deeply learned autoencoders. In *Pacific Symposium on Biocomputing 2017*, pages 207–218. World Scientific, 2017.

R. M. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *ICDM*, pages 43–52, 2007.

G. E. P. Box and G. Jenkins. *Time Series Analysis, Forecasting and Control*. Holden-Day, Inc., USA, 1990. ISBN 0816211043.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Z. Chen and A. Cichocki. Nonnegative matrix factorization with temporal smoothness and/or spatial decorrelation constraints. *Laboratory for Advanced Brain Signal Processing, RIKEN, Tech. Rep*, 2005.

M. Du, F. Li, G. Zheng, and V. Srikumar. Deeplog: Anomaly detection and diagnosis from system logs through deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, page 1285–1298, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349468. doi: 10.1145/3133956.3134015. URL https://doi.org/10.1145/3133956.3134015.

Y. Feng, J. Xiao, Y. Zhuang, X. Yang, J. J. Zhang, and R. Song. Exploiting temporal stability and low-rank structure for motion capture data refinement. *Information Sciences*, pages 777–793, 2014.

P. Filonov, A. Lavrentyev, and A. Vorontsov. Multivariate industrial time series with cyber-attack simulation: Fault detection using an lstm-based predictive data model. *CoRR*, abs/1612.06676, 2016. URL http://arxiv.org/abs/1612.06676.

K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, KDD '18, page 387–395, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3219845. URL https://doi.org/10.1145/3219819.3219845.

R. E. Kalman. A new approach to linear filtering and prediction problems. 1960.

Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *SIGKDD*, pages 426–434, 2008.

K.-H. Lai, D. Zha, G. Wang, J. Xu, Y. Zhao, D. Kumar, Y. Chen, P. Zumkhawaka, M. Wan, D. Martinez, and X. Hu. Tods: An automated time series outlier detection system. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(18):16060–16062, May 2021.

F. T. Liu, K. M. Ting, and Z. Zhou. Isolation forest. In *ICDM*, pages 413–422, 2008.

Y. Liu, Z. Li, C. Zhou, Y. Jiang, J. Sun, M. Wang, and X. He. Generative adversarial active learning for unsupervised outlier detection. *CoRR*, abs/1809.10816, 2018. URL http://arxiv.org/abs/1809.10816.

P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection. *CoRR*, abs/1607.00148, 2016a. URL http://arxiv.org/abs/1607.00148.

P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection. *ArXiv*, 2016b.

A. K. Nandi and H. Ahmed. *Frequency Domain Analysis*, pages 63–77. 2019. doi: 10.1002/9781119544678.ch4.

S. V. Narasimhan, N. Basumallick, and S. Veena. *Introduction to Wavelet Transform: A Signal Processing Approach*. Alpha Science International, Ltd, 1st edition, 2011. ISBN 1842656295.

D. Park, Y. Hoshi, and C. C. Kemp. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *CoRR*, abs/1711.00614, 2017. URL http://arxiv.org/abs/1711.00614.

D. Park, H. Kim, Y. Hoshi, Z. Erickson, A. Kapusta, and C. C. Kemp. A multimodal execution monitor with anomaly classification for robot-assisted feeding. In *2017 IEEE/RSJ International*

Conference on Intelligent Robots and Systems (IROS), pages 5406–5413, 2017. doi: 10.1109/IROS.2017.8206437.

H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, and Q. Zhang. Time-series anomaly detection service at microsoft. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, KDD '19, page 3009–3017, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330680. URL `https://doi.org/10.1145/3292500.3330680`.

A. Rodriguez, D. Bourne, M. T. Mason, G. F. Rossano, and J. Wang. Failure detection in assembly: Force signature analysis. In *Proceedings of IEEE Conference on Automation Science and Engineering (CASE 2010)*, pages 210 – 215, August 2010.

J. Ronald Eastman, F. Sangermano, B. Ghimire, H. Zhu, H. Chen, N. Neeti, Y. Cai, E. A. Machado, and S. C. Crema. Seasonal trend analysis of image time series. *Int. J. Remote Sens.*, 30(10): 2721–2726, Jan. 2009. ISSN 0143-1161. doi: 10.1080/01431160902755338. URL `https://doi.org/10.1080/01431160902755338`.

H. Sak, A. W. Senior, and F. Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. 2014.

B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt. Support vector method for novelty detection. In *NeurIPS*, pages 582–588, 2000.

R. Sen, H.-F. Yu, and I. S. Dhillon. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. In *NeurIPS*, pages 4837–4846, 2019.

J. H. Stock and M. W. Watson. Vector autoregressions. *Journal of Economic perspectives*, pages 101–115, 2001.

J. Ting, E. Theodorou, and S. Schaal. A kalman filter for robust outlier detection. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1514–1519, 2007. doi: 10.1109/IROS.2007.4399158.

D. Wang, H. Lian, Y. Zheng, and G. Li. High-dimensional vector autoregressive time series modeling via tensor decomposition. *ArXiv*, 2019.

Y.-X. Wang and Y.-J. Zhang. Nonnegative matrix factorization: A comprehensive review. *IEEE Transactions on Knowledge and Data Engineering*, pages 1336–1353, 2012.

G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, USA, 1995.

M. West and J. Harrison. *Bayesian forecasting and dynamic models*. Springer Science & Business Media, 2006.

L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SIAM SDM*, pages 211–222, 2010.

C.-C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. Keogh. Matrix profile i: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 1317–1322, 2016. doi: 10.1109/ICDM.2016.0179.

H.-F. Yu, N. Rao, and I. S. Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. In *NeurIPS*, pages 847–855, 2016.

Y. Zhang, M. Roughan, W. Willinger, and L. Qiu. Spatio-temporal compressive sensing and internet traffic matrices. In *SIGCOMM*, pages 267–278, 2009.

B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=BJJLHbb0-`.

APPENDIX A

EXPERIMENT DETAILS FOR ATMF

In this section, the details of the experiments is provided, including hyper-parameter settings, implementation and training details, as well as the comprehensive experiment results on the anomaly detection task.

A.1 Time Series Classical Tasks

For fair comparison, the lags of all algorithms was aligned, and other hyperparameters was tuned for each algorithm until it reaches the convergence. For Electricity and traffic {1,2,3,...24} was used as lag set and for wiki dataset {1,2,3,..14} was used as lag set. For MF algorithms, various latent dimensions {40,50,60,80,100,200,400,500} was tried and the one with the best performance was picked for each algorithm. The detail settings for each algorithm on the time series forecasting task is listed as follows:

- *AR(1)*: Linear regression was adopted for implementing the algorithm, and the L2-norm was used to learn the temporal correlations.

- *RNN-LSTM*: The 2 layers LSTM model was adopted and the hidden size with best performance was selected from the set {40,50,60,80,100,200,400,500}. The learning rate was tuned from set {0.1,0.01,0.001,0.0001}. Dropout rate was varied from the set {0.0,0.3,0.5}.

- *SVD-AR(1)*: The rank-k approximation $Y = USV^T$ is first obtained by SVD. After setting $F = US$ and $X = V^T$, AR(1) model was used as mentioned above to learn temporal correlations among $X$and the value was estimated using $US_kV'^T$, where S_k is k - dimensional approximation of S i.e. top k row of S and $V^{'}$ is estimated value of V obtained using AR(1). k was tuned from our latent dimension set and the one with best results was picked.

- *TRMF*: For electricity and traffic the Hyperparameters provided in the TRMF code was used and for wikipedia the hyperparameters were tuned from 0.000001 to 10 and picked the best results. The results are reported for following settings : Electricity (latent dimension = 60 , λ_w = 0.5, λ_x = 125, λ_f = 2 , learning rate = 0.0001) , traffic (latent dimension = 40 , λ_w = 2, λ_x = 625, λ_f = 0.5 , learning rate = 0.0001) and for wiki (latent dimension = 500 , λ_w = 0.00002, λ_x = 0.00625, λ_f = 0.000005 , learning rate = 0.0001)

- *ATMF*: The learning rate , regularization rate and latent dimension was tuned . Results were reported for following settings : Electricity (latent dimension = 60 , learning rate = 0.0001 , regularization rate = 1,window size = 24) , traffic (latent dimension = 40 , learning rate = 0.0001 , regularization rate = 10 , window size = 24) and wiki (latent dimension = 50 , learning rate = 0.0005 , regularization rate = 1 , window size = 14)

The detail settings on the missing data imputation task is listed as follows:

- *AE*: Keras default adam's optimizer's was used and learning rate was tuned from the set {0.1,0.01,0.001} and dropout from set {0,0.2,0.5}. The size of hidden dimension was fixed based on latent dimension used in MF methods, i.e. the (2 * latent dimension, latent dimension, latent dimension, 2 * latent dimension).

- *TRMF*: TRMF python code was used for this part of experiment . All hyperparameters was tuned from the set {0.00001,0.0001,0.001,0.1,1,10,100} and the best one was picked . The results are reported for following settings : Electricity (latent dimension = 60 , λ_x = 0.1, λ_f = 0.1, λ_w = 0.1 , learning rate = 0.0001) , traffic (latent dimension = 40 , λ_x = 0.1, λ_f = 0.1, λ_w = 0.1 , learning rate = 0.0001) and wiki dataset (latent dimension = 500 , λ_x = 0.01, λ_f = 0.01,λ_w = 0.01 , learning rate = 0.00001).

- *ATMF*: The learning rate , regularization rate and latent dimension were tuned . Results were reported for following settings : Electricity (latent dimension = 60 , learning rate = 0.0004 , regularization rate = 1,window size = 24) , traffic (latent dimension = 40 , learning rate = 0.0003 , regularization rate = 10 ,window size = 24) and wiki (latent dimension = 50 , learning rate = 0.0005 , regularization rate = 10 ,window size = 14)

A.2 Anomaly Detection

For fair comparison, the contamination ratio of all algorithms was aligned to 0.001 and 0.05 for Yahoo and SMD dataset respectively. The criteria to select the contamination ratio is corresponding to the anomaly ratio of each dataset. The hyperparamter of TRMF and ATMF were varied, trying to reach training error noraml deviation of 0.3 or convergence of training error upto to 3 deicmal places. For LSTM-ED ,the hyperparamters were varied and the combination with least training error was picked. For LSTM-ED and MF models the lag set was kept fixed as {1,2,3,..,20} and {1,2,3,....,100} for Yahoo A1 and SMD respectively. The detail settings on the anomaly detection task is listed as follows:

IForest: All features from data was used and the maximum estimator was fixed as 100.

OCSVM: Default parameters ie. kernel = 'rbf', degree=3, gamma='1/(number of features)', coef0=0.0, tol=1e-3, nu=0.5 was used for this experiment.

LSTM-ED: Various hidden dimension was tried from the set {5,10,20,40,50,100,200}. For optimizer's parameter Adam's default parameters was used.

TRMF: Results for following setting were reported : Yahoo(latent factors = 10, learning rate = 0.001, λ_f = 0.1, λ_x = 0.1, λ_w = 0.1) and SMD(latent factors = 10, learning rate = 0.0001, λ_f = 0.1, λ_x = 0.1, λ_w = 0.1)

ATMF: Results for following setting were reported : Yahoo(latent factors = 5, learning rate = 0.001, regularization rate = 0.01, window size = 20) and SMD (latent factors = 20, learning rate = 0.0001, regularization rate = 0.0001, window size = 100)

A.2.1 Implementation Details

The implementation details including the adopted libraries/packages are listed, the implementation methods as follows:

- *AR (1)*: This baseline is implemented by the linear regression of scikit-learn. The implementation is provided in the following GitHub repository:

  https://github.com/SemenovAlex/trmf/blob/master/Forecast.py

- *IForest*: The implementation in PyOD was adopted (https://github.com/yzhao062/pyod/blob/master/pyod/models/iforest.py).
- *OCSVM*: The implementation in PyOD was adopted (https://github.com/yzhao062/pyod/blob/master/pyod/models/ocsvm.py).
- *LSTM-ED*: The implementation in PyODDS was adopted (https://github.com/datamllab/pyodds/blob/master/pyodds/ algo/lstmencdec.py).
- *RNN-LSRM*: This baseline was implemented with Keras package (https://keras.io/).
- *SVD-AR*: The SVD was manually implemented and was combined with the aforementioned AR (1) for this baseline.
- *TRMF*: The implementation on GitHub was adopted (https://github.com/SemenovAlex/trmf/blob/master/trmf.py)

A.3 Comprehensive Results for Anomaly Detection Task

In this section the comprehensive results on each of the time series in the two real-world datasets and additional analysis for the anomaly detection task is provided

A.4 Additional analysis

There are mainly two types of anomalies presented in both datasets. First, global anomalies which values are far from the entire dataset. Second, contextual anomalies that values significantly deviates from their context (neighboring) points. In the case of contextual anomalies, the same value may not be considered as an anomaly if it occurs in other context (neighborhood). Global anomalies identification is more suitable for cluster based algorithms like IForest and OCSVM, most MSE penalty based algorithm's convergence might become biased towards these because of it's nature of giving large penalty to large residuals. Contextual anomalies are more suitable for models which learns temporal correlation among data points. As can seen from the results ATMF because of its MAE based penalty functions and temporal neighbourhood model, it is able to detect both kind of anomalies and thus outperforms the baselines.

In case of Yahoo-A1 , the dataset marked with '*' has no annotated anomaly point. In the real_48 and real_26 time series, since the deviation of the annotated anomalies is smaller than the deviation of all other normal points, cluster-based methods which detect the anomalies with higher deviation will fail the task. On the other hand, the anomalies of real_48 and real_26 are the change points, which means after the anomalies, the value of the time series drastically drop to values with absolute z-score less than 1. Furthermore, the seasonality of the two datasets are not stable, both magnitude and length of each period varies a lot in the two datasets. This phenomenon leads to the methods with single lag set, i.e. both *TRMF* and *ATMF* fail to properly model the temporal correlations for the anomalies and therefore not able to detect the anomalies.

Yahoo-A1	OCSVM	IForest	LSTM-ED	TRMF	ATMF
No.	(Pre./Rec./F1)	(Pre./Rec./F1)	(Pre./Rec./F1)	(Pre./Rec./F1)	(Pre./Rec./F1)
real_1	1/1/1	1/1/1	0.5/0.5/0.5	0/0/0	1/1/1
real_2	1/0.125/0.222	1/0.125/0.222	1/0.125/0.222	0.5/0.062/0.111	1/0.125/0.222
real_3	1/0.133/0.235	1/0.133/0.235	1/0.133/0.235	0.5/0.066/0.118	1/0.133/0.235
real_4	1/0.4/0.571	1/0.4/0.571	1/0.4/0.571	0/0/0	1/0.4/0.571
real_5	1/1/1	1/1/1	0.5/0.5/0.5	0.5/0.5/0.5	0.5/0.5/0.5
real_6	1/0.25/0.4	1/0.25/0.4	0.5/0.125/0.2	0/0/0	1/0.25/0.4
real_7	0/0/0	0/0/0	0.5/0.0164/0.0317	0/0/0	1/0.033/0.0635
real_8	0.5/0.1/0.167	1/0.1/0.183	0.5/0.1/0.167	0/0/0	1/0.2/0.333
real_9	1/0.25/0.4	1/0.25/0.4	1/0.25/0.4	0.5/0.125/0.2	1/0.25/0.4
real_10	1/0.154/0.267	1/0.154/0.267	0/0/0	0.5/0.077/0.133	1/0.154/0.267
real_11	1/0.105/0.190	1/0.105/0.190	1/0.105/0.190	0/0/0	1/0.105/0.190
real_12	1/0.667/0.8	1/0.667/0.8	1/0.667/0.8	0.5/0.333/0.4	1/0.667/0.8
real_13	1/0.167/0.286	1/0.167/0.286	1/0.167/0.286	0/0/0	1/0.167/0.286
real_14	0.5/0.5/0.5	0.5/0.5/0.5	0.5/0.5/0.5	0/0/0	0.5/0.5/0.5
real_15	1/0.2/0.333	1/0.2/0.333	1/0.2/0.333	0/0/0	1/0.2/0.333
real_16	1/0.67/0.8	1/0.67/0.8	0.5/0.33/0.4	0/0/0	1/0.67/0.8
real_17	1/0.009/0.017	1/0.009/0.017	1/0.009/0.017	1/0.009/0.017	1/0.009/0.017
real_18	1/0.667/0.8	1/0.667/0.8	1/0.667/0.8	0/0/0	0.5/0.333/0.4
real_19	1/0.009/0.017	1/0.008/0.017	1/0.008/0.017	0.5/0.004/0.008	0.5/0.004/0.008
real_20	0.5/0.03/0.057	0.5/0.03/0.057	1/0.06/0.114	0.5/0.03/0.057	1/0.06/0.114
real_21	1/0.333/0.5	1/0.333/0.5	0.5/0.167/0.25	0.5/0.167/0.25	1/0.333/0.5
real_22	1/0.032/0.062	1/0.032/0.062	1/0.032/0.062	0.5/0.016/0.031	1/0.032/0.062
real_23	0.5/0.052/0.095	0/0/0	0.5/0.052/0.095	0.5/0.052/0.095	1/0.105/0.191
real_24	1/0/125/0.222	1/0/125/0.222	1/0/125/0.222	1/0/125/0.222	1/0/125/0.222
real_25	1/0.046/0.089	1/0.046/0.089	0.5/0.023/0.044	0/0/0	1/0.046/0.089
real_26	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
real_27	0.5/0.5/0.5	0.5/0.5/0.5	0.5/0.5/0.5	0/0/0	0.5/0.5/0.5
real_28	0/0/0	0/0/0	1/0.024/0.047	0/0/0	1/0.024/0.047
real_29	0.5/0.143/0.222	0.5/0.143/0.222	0.5/0.143/0.222	0/0/0	1/0.286/0.422
real_30	1/0.222/0.364	1/0.222/0.364	1/0.222/0.364	0/0/0	1/0.222/0.364
real_31	1/0.083/0.154	1/0.083/0.154	1/0.083/0.154	0/0/0	1/0.083/0.154
real_32	1/0.042/0.082	1/0.042/0.082	0.5/0.021/0.040 real_	0.5/0.021/0.040	1/0.042/0.082
real_33	1/1/1	1/1/1	1/1/1	0.5/0.5/0.5	1/1/1
real_34	1/0.286/0.444	1/0.286/0.444	1/0.286/0.444	0/0/0	1/0.286/0.444
real_35*	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
real_36	1/0.4/0.571	1/0.4/0.571	1/0.4/0.571	1/0.4/0.571	1/0.4/0.571

Table A.1: Experiment details of Anomaly Detection on Yahoo Dataset.

Yahoo-A1	OCSVM	IForest	LSTM-ED	TRMF	ATMF
No.	(Pre./Rec./F1)	(Pre./Rec./F1)	(Pre./Rec./F1)	(Pre./Rec./F1)	(Pre./Rec./F1)
real_37	1/0.059/0.111	1/0.059/0.111	0.5/0.029/0.055	0/0/0	1/0.059/0.111
real_38	0.5/0.111/0.182	0.5/0.111/0.182	1/0.222/0.364	0/0/0	1/0.222/0.364
real_39	1/0.2/0.333	1/0.2/0.333	1/0.2/0.333	0/0/0	1/0.2/0.333
real_40	0/0/0	0/0/0	0.5/0.012/0.024	0/0/0	0/0/0
real_41	1/0.667/0.8	1/0.667/0.8	1/0.667/0.8	0.5/0.333/0.4	1/0.667/0.8
real_42	1/0.045/0.087	1/0.045/0.087	1/0.045/0.087	0.5/0.022/0.044	1/0.045/0.087
real_43	1/0.062/0.118	1/0.062/0.118	1/0.062/0.118	0/0/0	1/0.062/0.118
real_44	1/0.222/0.364	1/0.222/0.364	1/0.222/0.364	0/0/0	1/0.222/0.364
real_45	0.5/1/0.667	0.5/1/0.667	0.5/1/0.667	0.5/1/0.667	0.5/1/0.667
real_46	0/0/0	0/0/0	1/0.018/0.036	0/0/0	0.5/0.09/0.018
real_47	0/0/0	0/0/0	1/0.2/0.333	0/0/0	0.5/0.1/0.167
real_48	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
real_49	1/0.667/0.8	1/0.667/0.8	0/0/0	0/0/0	0.5/0.333/0.4
real_50	1/0.286/0.444	1/0.286/0.444	1/0.286/0.444	0/0/0	1/0.286/0.444
real_51	0/0/0	0.5/0.25/0.333	1/0.5/0.667	0.5/0.25/0.333	0.5/0.25/0.333
real_52	0.5/0.091/0.154	0.5/0.091/0.154	1/1.182/0.308	0/0/0	1/1.182/0.308
real_53	1/0.118/0.210	1/0.118/0.210	1/0.118/0.210	0.5/0.059/0.105	1/0.118/0.210
real_54	1/0.25/0.4	1/0.25/0.4	1/0.25/0.4	0/0/0	1/0.25/0.4
real_55	0.5/0.2/0.286	0.5/0.2/0.286	1/0.4/0.571	0/0/0	0.5/0.2/0.286
real_56	0.5/0.2/0.286	0.5/0.2/0.286	1/0.4/0.571	0/0/0	1/0.4/0.571
real_57	0/0/0	0/0/0	1/0.667/0.8	0/0/0	0/0/0
real_58	1/0.046/0.089	1/0.046/0.089	0.5/0.023/0.044	0/0/0	1/0.046/0.089
real_59*	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
real_60	1/0.125/0.222	1/0.062/0.118	0.5/0.062/0.111	0.5/0.062/0.111	1/0.125/0.222
real_61	0.5/0.042/0.077	0.5/0.042/0.077	0/0/0	0/0/0	0.5/0.042/0.077
real_62	1/0.143/0.25	1/0.143/0.25	1/0.143/0.25	0/0/0	1/0.143/0.25
real_63	1/0.25/0.4	1/0.25/0.4	0/0/0	0/0/0	0.5/0.125/0.2
real_64*	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
real_65	1/0.12/0.21	1/0.12/0.21	1/0.12/0.21	1/0.12/0.21	1/0.12/0.21
real_66	1/0.095/0.174	1/0.095/0.174	0.5/0.047/0.087	0/0/0	1/0.095/0.174
real_67	1/0.087/0.16	1/0.087/0.16	1/0.087/0.16	0.5/0.043/0.08	1/0.087/0.16

Table A.2: Continued Experiment details of Anomaly Detection on Yahoo Dataset.

SMD	IForest	OCSVM	LSTM-ED	TRMF	ATMF
No.	(Pre./Rec./F1)	(Pre./Rec./F1)	(Pre./Rec./F1)	(Pre./Rec./F1)	(Pre./Rec./F1)
1	0.581/0.307/0.402	0.57/0.301/0.394	0.331/0.175/0.229	0.643/0.34/0.445	0.674/0.356/0.466
2	0.231/0.506/0.317	0.239/0.522/0.328	0.233/0.509/0.32	0.224/0.491/0.308	0.223/0.487/0.306
3	0.185/0.269/0.22	0.18/0.261/0.213	0.345/0.501/0.408	0.223/0.323/0.264	0.377/0.547/0.446
4	0.154/0.254/0.192	0.176/0.29/0.219	0.176/0.29/0.219	0.226/0.372/0.281	0.291/0.479/0.362
5	0.055/0.65/0.101	0.059/0.7/0.109	0.056/0.66/0.103	0.06/0.71/0.11	0.068/0.81/0.126
6	0.157/0.244/0.191	0.197/0.305/0.239	0.165/0.256/0.2	0.208/0.324/0.254	0.377/0.586/0.459
7	0.228/0.231/0.229	0.235/0.238/0.237	0.23/0.232/0.231	0.23/0.233/0.232	0.365/0.37/0.368
8	0.22/0.092/0.13	0.197/0.082/0.116	0.153/0.064/0.09	0.171/0.072/0.101	0.166/0.07/0.098
9	0.186/0.818/0.303	0.184/0.81/0.3	0.186/0.818/0.303	0.186/0.818/0.303	0.197/0.866/0.32
10	0.567/0.397/0.467	0.58/0.406/0.477	0.354/0.247/0.291	0.619/0.433/0.51	0.66/0.462/0.543
11	0.205/0.248/0.224	0.333/0.403/0.365	0.194/0.235/0.212	0.403/0.488/0.442	0.427/0.516/0.467
12	0.138/0.467/0.213	0.12/0.408/0.186	0.12/0.406/0.185	0.12/0.408/0.186	0.155/0.526/0.24
13	0.132/0.969/0.232	0.132/0.969/0.232	0.132/0.975/0.233	0.132/0.969/0.232	0.134/0.988/0.236
14	0.546/0.447/0.491	0.543/0.445/0.489	0.36/0.295/0.324	0.525/0.43/0.472	0.513/0.42/0.462
15	0.103/0.481/0.17	0.151/0.705/0.249	0.126/0.588/0.208	0.148/0.692/0.244	0.172/0.802/0.283
16	0.072/0.077/0.074	0.073/0.078/0.075	0.06/0.064/0.062	0.079/0.085/0.082	0.081/0.087/0.084
17	0.132/0.247/0.172	0.175/0.329/0.229	0.222/0.416/0.289	0.211/0.396/0.275	0.28/0.525/0.365
18	0.225/0.627/0.331	0.254/0.707/0.374	0.298/0.829/0.438	0.234/0.65/0.344	0.244/0.678/0.359
19	0.126/0.417/0.194	0.13/0.429/0.199	0.133/0.44/0.204	0.098/0.325/0.151	0.173/0.574/0.266
20	0.403/0.422/0.412	0.331/0.347/0.339	0.207/0.217/0.212	0.228/0.239/0.233	0.325/0.341/0.333
21	0.079/0.752/0.144	0.142/0.673/0.235	0.132/0.627/0.219	0.145/0.686/0.239	0.154/0.729/0.254
22	0.684/0.774/0.726	0.7/0.793/0.744	0.651/0.737/0.692	0.654/0.74/0.694	0.613/0.693/0.651
23	0.067/0.485/0.118	0.063/0.46/0.111	0.084/0.606/0.147	0.072/0.52/0.126	0.105/0.758/0.184

Table A.3: Experiment details of Anomaly Detection on Sever Machine Dataset.